



VYSOKÁ ŠKOLA BÁŇSKÁ — TECHNICKÁ UNIVERZITA OSTRAVA  
EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Návrh racionalizace subsystému

Subsystem rationalisation design

Student: Zdeněk Jeřábek

Vedoucí bakalářské práce: Doc. Ing. Ludmila Kalužová, CSc.

Ostrava 2008

„Místopřísežně prohlašuji, že jsem celou práci včetně všech příloh vypracoval samostatně.

Rád bych touto cestou poděkoval svému vedoucímu bakalářské práce paní Doc. Ing. Ludmile Kalužové, CSc. za její cenné připomínky, ochotu a mimořádnou trpělivost při vedení mé práce a zároveň také paní Ing. Macháčkové, za poskytnutí materiálů a informací a za spolupráci a cenné rady při realizaci práce.“

*25. dubna 2008*

# Obsah

<b>1</b>	<b>Teoretická východiska pro řešení</b>	<b>5</b>
1.1	Databáze . . . . .	5
1.2	Systémy řízení báze dat . . . . .	6
1.3	Datový model . . . . .	6
1.4	Sémantické modelování . . . . .	7
1.4.1	Identifikace vstupních datových požadavků . . . . .	8
1.4.2	Specifikace datových objektů a jejich charakteristik . . . . .	8
1.4.3	Korekce struktury datových objektů . . . . .	8
1.5	Konceptuální modelování . . . . .	8
1.5.1	Vymezení struktury entit . . . . .	9
1.5.2	Přiřazení primárních klíčů entitám . . . . .	10
1.5.3	Definování vztahů . . . . .	10
1.5.4	Integrace dílčích částí modelu . . . . .	10
1.6	Logické relační modelování . . . . .	10
1.6.1	Vytvoření soustavy předběžných relací . . . . .	11
1.6.2	Přiřazení zbývajících atributů . . . . .	11
1.6.3	Revize konceptuálního modelu . . . . .	11
1.6.4	Normalizace modelu . . . . .	11
1.6.5	Specifikace domén . . . . .	12
1.7	Základy MS Access 2003 . . . . .	12
1.7.1	Upravení výsledné aplikace . . . . .	16
<b>2</b>	<b>Analýza a zhodnocení současného stavu řešené problematiky</b>	<b>19</b>
<b>3</b>	<b>Návrh racionalizovaného řešení</b>	<b>20</b>
3.1	Sémantické modelování . . . . .	20
3.1.1	Identifikace vstupních datových požadavků . . . . .	20
3.1.2	Specifikace datových objektů a jejich charakteristik . . . . .	21

3.1.3	Korekce struktury datových objektů + struktura entit pro konceptuální model . . . . .	21
3.2	Konceptuální modelování . . . . .	21
3.3	Logické relační modelování . . . . .	22
3.3.1	Vytvoření soustavy předběžných relací . . . . .	22
3.3.2	Přiřazení zbývajících atributů . . . . .	22
3.3.3	Specifikace domén . . . . .	23
3.4	Dekompozice funkcí . . . . .	26
3.5	Procesní popis . . . . .	28
<b>4</b>	<b>Zhodnocení výsledků zpracovaného řešení</b>	<b>33</b>
<b>5</b>	<b>Závěr</b>	<b>34</b>
	<b>Seznam použité literatury</b>	<b>35</b>
	<b>Seznam zkratk</b>	<b>36</b>
	<b>Prohlášení o využití výsledků bakalářské práce</b>	<b>37</b>

# Úvod

Dnešní doba si žádá rychlé, přesné a relevantní informace. Za účelem efektivního zpracování těchto informací vznikají tzv. informační systémy.

*Informační systém je soubor lidí, technických prostředků a metod (programů), zabezpečujících sběr, přenos, zpracování, uchování dat, za účelem prezentace informací pro potřeby uživatelů činných v systémech řízení. [3]*

V oddělení zahraničních styků na VŠB-TU Ekonomické fakultě vznikla potřeba racionalizovat zpracování informací, týkajících se zahraničních studentů zde studujících pod záštitou programu Erasmus, CEEPUS, nebo na základě bilaterální dohody mezi školami.

Cílem práce je tedy navrhnout a vytvořit aplikaci, která usnadní a zefektivní procesy spojené se zpracováním dat o příchozích zahraničních studentech, jejich další evidencí (přihlašování na předměty apod.) a následné vyřazení z databáze a přesun do archivu.

# Kapitola 1

## Teoretická východiska pro řešení

### 1.1 Databáze

Podle [2] zní definice databáze:

*Databáze je souhrnem souvisejících dat uložených bez redundancí a sloužících řadě aplikací.*

**K základním vlastnostem databází patří:**

**Minimalizace redundance dat** — úplné odstranění redundance (nadbytečnosti) není mnohdy možné, a tak se v některých databázích nalézají redundantní údaje, které jsou akceptovány například za účelem zrychlení běhu operací. Tyto údaje musí být důsledně aktualizovány aby nedošlo k nejednotnosti v datech.

**Integrita databáze** se týká redundance dat, popř. špatných údajů zapsaných do databáze (např. nesmyslné datum narození uvedené u pracovníka firmy). Tomuto předchází správně napsaná databáze, která svými mechanismy kontroluje správnost těchto údajů (např. rozsah platných dat narození).

**Nezávislost dat** na aplikačních programech umožnily tzv. systémy řízení báze dat (database management system), které oddělují aplikační programy od samotných dat.

**Sdílení dat.** Pro nově vzniklé aplikace se nemusí tvořit nové datové struktury. Takto se celý systém stává pružnější pro případné změny, popř. rozšíření.

**Pružnost** — uživatel není odkázán jen na určité aplikace, ale může využít dotazovacích jazyků pro specifické potřeby výběrů dat.

**Ochrana dat** z pohledu přístupů jednotlivých uživatelů. Data bývají často soustředována do jednoho umístění, a tak může hrozit jejich únik a případné zneužití. Proto by měl systém umožnit přidělování práv přístupu jednotlivým uživatelům databáze (např. pracovník účtárny firmy bude mít přístup pouze k relevantním údajům týkající se účetnictví) a omezit manipulaci s daty — udělování práv k zápisu, mazání, editaci údajů apod.

## 1.2 Systémy řízení báze dat

S databázemi velice úzce souvisí pojem *Systémy řízení báze dat* (*Database Management System*). Jak už bylo uvedeno, SŘBD oddělují aplikační programy od samotných dat. Tyto systémy provádí operace s databázemi. K základním funkcím patří: *definice struktury databáze* (která je uložena v tzv. slovníku dat) a *naplnění / aktualizace / výběr* dat.

Systémy řízení báze dat se skládají ze čtyř částí. Každá část má na starost specifickou oblast operací.

1. Překladače databázových jazyků „překládají“ příkazy pro další zpracování výkonnými programy.
2. Programy pro práci se slovníkem dat provádí operace se slovníkem dat (např. úpravy popisu struktury databáze).
3. Výkonné programy umožňují naplnění / aktualizaci / výběr dat, dále pak provádí ochranu, tj. neumožní přístup neoprávněným uživatelům, starají se o integritu databáze apod.
4. Služební programy tvoří statistiky o využívání dat, umožňují nestandardní naplnění dat apod.

## 1.3 Datový model

Profesor Kaluža [1] uvádí svou definici datového modelu:

*Datový model jako vyjádření datové struktury modelovaného informačního systému. Tento model je pak abstrakcí, odrazem reálného světa z pohledu designera realizujícího cíle, které má projekt dosáhnout.*



Způsobů tvorby datového modelu je více, ovšem profesor Kaluža [1] navrhnul tzv. *tříúrovňovou koncepci datového modelování*, které se v této práci přidržím.

Tato metoda logicky postupuje od obecnějšího slovního vyjádření (sémantické modelování), přes grafické vyjádření struktury modelu (konceptuální modelování), až po datový model závislý na konkrétní databázové koncepci (logické relační modelování).

#### **Tříúrovňová koncepce datového modelování:**

##### **- Sémantické modelování**

- Identifikace vstupních datových požadavků
- Specifikace datových objektů a jejich charakteristik
- Korekce struktury datových objektů

##### **- Konceptuální modelování**

- Vymezení struktury entit
- Přiřazení primárních klíčů entitám
- Definování vztahů
- Integrace dílčích částí modelu

##### **- Logické relační modelování**

- Vytvoření soustavy předběžných relací
- Přiřazení zbývajících atributů
- Revize konceptuálního modelu
- Normalizace modelu
- Specifikace domén

## **1.4 Sémantické modelování**

Tento proces má za úkol slovně zachytit celou část zkoumané reality, pro kterou je potřeba vytvořit datový model. Jako výsledek této etapy vznikne objektová struktura.

### **1.4.1 Identifikace vstupních datových požadavků**

Identifikace vstupních datových požadavků může proběhnout ve více formách. Mezi základní patří rozhovor s uživatelem, popř. anketa, analýza písemných pramenů a využití starých datových struktur jako podkladu pro nové.

Anketa se využívá spíše okrajově, pro svou neschopnost interakce s uživatelem. Proto ji lze využít spíše jako počáteční průzkum, tak aby si mohl designer lépe připravit otázky k samotnému rozhovoru s uživatelem.

Rozhovor by měl být konstruktivní a měl by sledovat stanovené cíle. Hlavním cílem je objasnění potřeb a požadavků uživatele a obsáhlé vymezení dané problematiky.

Analýzou písemných pramenů se designer snaží promítnout relevantní informace do datového modelu, ať už jde o legislativní normy, návody, nebo o různé formuláře, výkazy apod. Formuláře jsou klasickým příkladem převedení písemných pramenů do elektronické podoby.

Starší datové struktury nám mohou v některých případech posloužit jako podklad pro tvorbu nových. Může být ponechána stejná struktura, nebo si uživatel může přát úpravu modelu (např. přidat, či odebrat nějaký objekt).

### **1.4.2 Specifikace datových objektů a jejich charakteristik**

Rozborem údajů, které jsme zjistili identifikací vstupních datových požadavků, specifikujeme jednotlivé objekty (plus jejich charakteristiky). Získané údaje je vhodné pro přehlednost zaznamenat ve strukturovaném zápise.

### **1.4.3 Korekce struktury datových objektů**

V této dílčí části je nezbytné najít a eliminovat případné chyby v dosavadním popisu. Slovní forma totiž umožňuje nepřesnou specifikaci a různou interpretaci, tzn. že hrozí nebezpečí redundance, synonym, homonym apod.

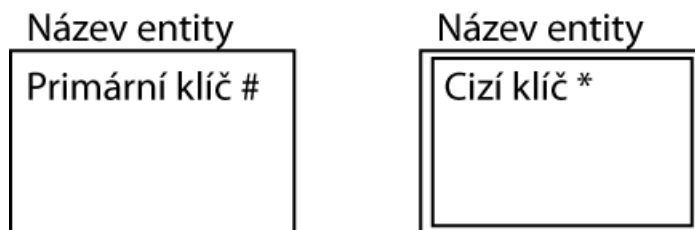
## **1.5 Konceptuální modelování**

Oblíbenou metodou konceptuálního modelování je E–R model, založený na grafické interpretaci datové struktury. Umožňuje poměrně snadné pochopení i ze strany uživatele. Proto se může uživatel stále podílet na vývoji tohoto modelu.

### 1.5.1 Vymezení struktury entit

Jednotlivé entity se pojmenují a graficky se vyznačí jejich správná struktura. Následuje výběr některých konstruktorů a jejich definice:

*Podle [1]: Entita — v E–R modelu vyjadřuje třídu objektů reálného světa. Graficky je entita vyjádřena obdélníkem s uvedením svého názvu, obrázek 1.1.*



Obr. 1.1: Entita, slabá entita, primární klíč a cizí klíč

*Atribut — je vlastnost entity. Např. entita osoba má atributy jméno, datum narození apod.*

*Primární klíč — je unikátní údaj, identifikující jednotlivé výskyty entity, který byl vybrán z kandidátních klíčů<sup>1</sup>. Grafické znázornění v E–R modelu je název primárního klíče a znak hash #, obrázek 1.1.*

*Cizí klíč — je klíč v entitě A, který je primárním klíčem v entitě B. Grafické znázornění spočívá ve jméně tohoto klíče doplněným znakově \*, obrázek 1.1.*

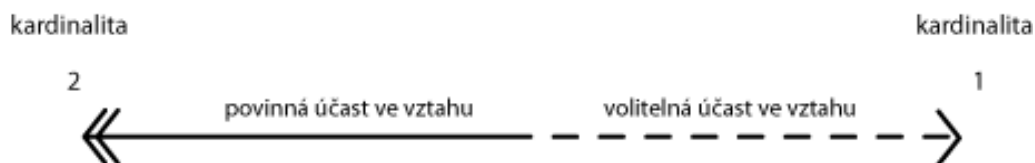
*Slabá entita — mající cizí klíč. Graficky se znázorní dvojítm obdelníkem a svým názvem, obrázek 1.1.*

*Vztah<sup>2</sup> — se zobrazí čárou spojující entity a to buď plnou (povinná účast entity ve vztahu), nebo přerušovanou (nepovinná účast entity), obrázek 1.2.*

*Složený atribut — názorná ukázka složeného atributu je např. adresa trvalého bydliště.*

<sup>1</sup>Podle [1] platí, že: kandidátní klíč — klíč jednoznačně identifikující výskyty dané entity

<sup>2</sup>Vlastnosti vztahu: *stupeň* (kolik entit je v jednom vztahu), *kardinalita* (počet výskytů entit ve vztahu k sobě navzájem) a *volitelnost* (povinná nebo volitelná přítomnost entity ve vztahu).



Obr. 1.2: Grafické znázornění vztahů

### 1.5.2 Přiřazení primárních klíčů entitám

Jednotlivým entitám přidělíme kandidátní klíče a z nich vybereme ke každé entitě klíč primární. Do grafického znázornění entity vložíme název primárního klíče s příslušným grafickým označením.

### 1.5.3 Definování vztahů

V této fázi se designer snaží rozpoznat existující vztahy mezi entitami a graficky je vyjádřit v E–R modelu. Tj. určí stupeň vztahu, kardinalitu a volitelnost.

Je možné různé znázornění entit a jejich vzájemných vztahů v E–R modelu. Zodpovědnost leží na designerovi, který využívá zkušeností, tak aby co nejlépe charakterizoval danou realitu v E–R modelu (např. s ohledem na rychlost a účinnost výsledného řešení).

### 1.5.4 Integrace dílčích částí modelu

Větší projekty bývají vytvářeny odděleně, po částech. V závěru je třeba tyto dílčí části spojit do jednoho celku. Profesor Kaluža [1] doporučuje tzv. binární přístup, kdy se postupně integrují dvojce dílčích částí.

Při spojování může docházet k defektům v modelu, vzniklých právě v důsledku oddělené tvorby. Tyto defekty mohou být např. synonyma, homonyma, nesoulad klíčů. Řešení může být přejmenování konstruktorů (synonyma, homonyma), nalezení jiného primárního klíče, nebo změna struktury (při nesouladu klíčů).

## 1.6 Logické relační modelování

V této fázi dochází k převedení E–R modelu do soustavy relací. Přechází se z grafické do textové formy. Relace je v podstatě tabulka, kde sloupce znamenají atributy

a řádky jsou jednotlivé výskyty, tzv. *n*-tice. Přičemž každá *n*-tice je jednoznačně identifikovaná atributem / atributy — klíčem / klíči.

Relace je popsána jejím názvem<sup>3</sup>, v závorce za názvem následují klíče a zbylé atributy. Tento zápis je patrný z příkladu:

NAZEV\_RELACE(PRIM\_KLIC#, ATRIBUT1, ATRIBUT2, ...)

### 1.6.1 Vytvoření soustavy předběžných relací

Pomocí jména a kandidátních klíčů se znázorní relace. Vytvořený zápis (bez zbývajících atributů) je přehlednější.

### 1.6.2 Přiřazení zbývajících atributů

K relacím se přiřadí zbývající atributy, dosud nezahrnuté v předběžných relacích. Přiřazování by se mělo dít v souladu s primárními klíči daných relací, aby se ušetřily problémy při normalizaci modelu.

### 1.6.3 Revize konceptuálního modelu

Revize konceptuálního modelu proběhne po přiřazení zbývajících atributů a poznání možných dalších souvislostí a případném uznání nutnosti změn v modelu. Designer se rozhodne, jakým způsobem bude řešit nastalé nesrovnalosti a jestli je nutná úprava modelů.

### 1.6.4 Normalizace modelu

Ve této fázi podstoupí vytvořené relace normalizaci. Normalizací modelu se eliminují případné anomálie vzniklé např. nežádoucími funkčními závislostmi mezi atributy. Mezi základní normální formy patří první, druhá a třetí normální forma a tzv. Boyce–Coddova normální forma (BCNF)<sup>4</sup>, která je mírně silnější než třetí normální forma. Existují i vyšší normální formy, ale tyto již nejsou tak běžně používány. Následuje definice BCNF podle [1]:

*Relace je v Boyce-Coddově normální formě právě tehdy, jestliže každý determinant funkční závislosti v relaci je zároveň kandidátním klíčem.*

<sup>3</sup>Důrazně se nedoporučuje používat diakritiku pro názvy relací a jejich atributy. Rovněž některé znaky nejsou povoleny (tyto zakázané znaky se většinou liší a je třeba prostudovat dokumentaci pro daný program).

<sup>4</sup>Podle profesora Kaluži, zpravidla stačí upravit model do Boyce–Coddovy normální formy.

### 1.6.5 Specifikace domén

Jednotlivým atributům je třeba určit vlastnosti. Tyto vlastnosti tvoří doménu, určující rámec hodnot, které je možné zadávat do atributu. Různé programy se mohou drobně odlišovat v těchto vlastnostech a jejich specifik, ale za základní lze považovat tyto:

- typ (zda se jedná o číslo, znak, datum, čas)
- délka (maximální povolený počet znaků)
- rozsah (meze daného atributu, např. data narození)
- jedinečnost (zda je, nebo není hodnota unikátní)
- null hodnoty (povolení / zakázání tzv. null hodnot)
- formát (např. způsob zápisu data narození, liší se v různých státech)
- přípustné hodnoty (povolené hodnoty)
- popis (jedná se o stručnou charakteristiku atributu)

## 1.7 Základy MS Access 2003

MS Access 2003<sup>5</sup> je RDBMS (Relational Database Management System), tedy relační systém řízení báze dat. Soubor s příponou *.mdb* je funkční RDBMS.

Mimo tradičních pojmů jako jsou atribut, nebo vztah, je podle [5] uváděn pojem *instance* — což znamená n-tice, tedy konkrétní záznam v tabulce.

Kromě přípony *.mdb* (sama funguje jako RDBMS), nabízí Access možnost vytvořit databázi s příponou *.adp* — tento soubor je ovládán Microsoft SQL Serverem. Tedy Microsoft SQL Server plní funkce RDBMS.

Významným prvkem kompatibility je podpora standardu ODBC (Open Database Connectivity). Podle [5] ODBC poskytuje standardní rozhraní do mnoha různých druhů databázových systémů. Access je tedy schopný pracovat i s jinými databázemi, podporujícími tento standard.

Access pro práci s daty využívá jazyka SQL (Structured Query Language), tedy standardizovaného dotazovacího jazyka. Pro tvorbu konkrétního dotazu lze využít grafický nástroj, který velmi usnadní práci, ovšem pro složitější dotazy tento nástroj nestačí a dotaz se musí psát ručně.

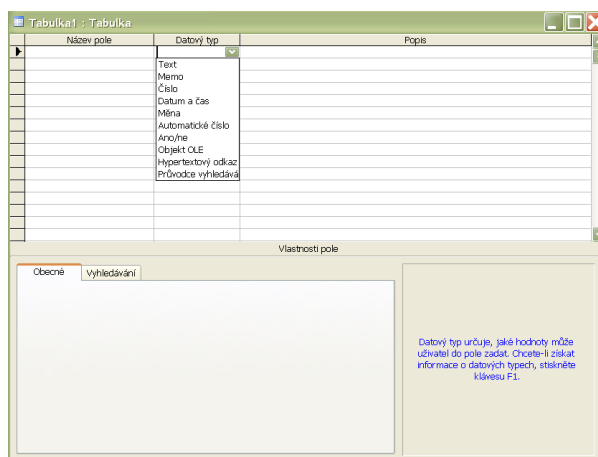
---

<sup>5</sup>Veškerá práce probíhá v Microsoft Access 2003, verze 11.0, aplikace VBA verze 6.05.

Tvorba databází probíhá za pomoci objektů MS Access, mezi tyto objekty patří:

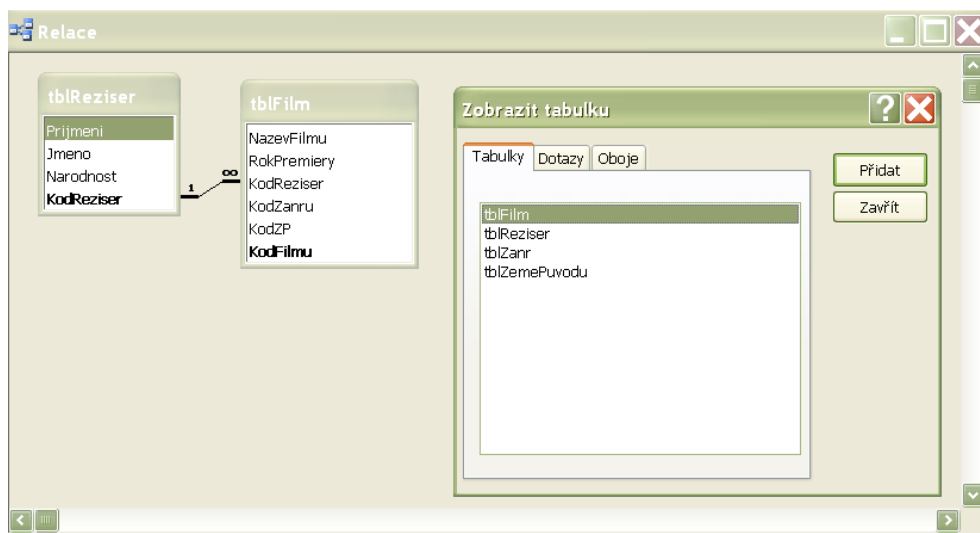
## Tabulky

Tabulky nesou uložená data a tvoří se v tzv. návrhovém zobrazení (obrázek 1.3). Na tomto obrázku jsou patrné i možnosti výběru datových typů.



Obr. 1.3: Návrhové zobrazení tvorby tabulky se zobrazením datových typů

K definici vztahů mezi relacemi se v MS Access používá grafická forma, viz obrázek 1.4. Tento grafický nástroj je využit k implementaci již vytvořeného datového modelu.



Obr. 1.4: Tvorba vztahů v MS Accessu

## Dotazy

Dotazy se mohou tvořit v návrhovém zobrazení, nebo psát přímo v jazyce SQL. Při tvorbě v návrhovém zobrazení Access umožní vybrat tabulky, kterých se týká daný dotaz.

V MS Access jsou dva druhy dotazů, které se liší svým použitím:

1. Výběrové dotazy — Vybírají data z tabulek, výběr lze omezit nastavením kritérií.
2. Akční dotazy — Provádí změny dat v tabulkách. Například:
  - Přidávací dotaz — Umí přesunout data do existující tabulky.
  - Odstraňovací dotaz — Odstraňuje n-tice z tabulky podle zadaných hodnot kritérií.

## Formuláře

Formuláře plní funkci větší uživatelské přívětivosti a funkčnosti. Nejsou závislé na struktuře tabulek. Data lze do formulářů vkládat, zobrazovat a přes formuláře se mohou také řídit určité operace. Do formulářů se umísťují ovládací prvky: rozbalovací seznamy, objekty OLE (obrázky, grafy, ...), tlačítka (vlození nového záznamu, uzavření formuláře, ...), textové pole apod. Tyto ovládací prvky lze ve formulářích téměř libovolně umísťovat a upravovat, například zda budou jen ke čtení, zda je lze aktualizovat, nebo jestli budou tyto prvky vidět.

Zajímavá je také možnost tisku formulářů. Většinou se k tisku používají sestavy, ale může nastat situace, kdy bude výhodnější, nebo vhodnější využít této funkce.

### **V programu MS Access 2003 lze vytvořit více druhů formulářů.**

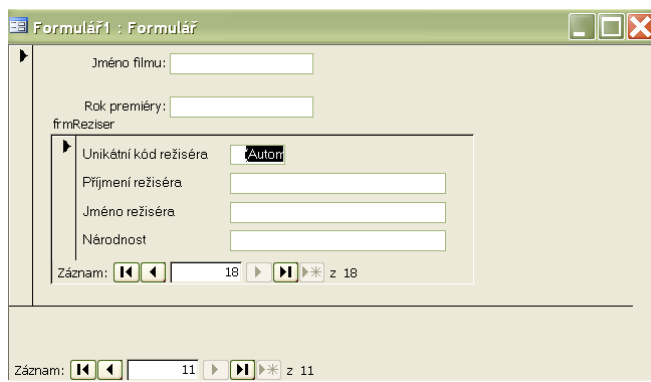
Mezi nejdůležitější a nejčastěji používané patří:

**Vícestránkové formuláře** — používají se, pokud je o určitém objektu mnoho informací. Při listování mezi záznamy se zobrazí vždy první stránka s údaji.

**Modální formuláře** — jestliže jsou aktivní, nelze vybrat jiné okno, tzn. že aplikace čeká od uživatele nějakou akci (potvrzení, vyhledání, uzavření, ...).

**Podformulář** — používá se zejména na vztahy typu 1 : n. Hlavní formulář je typu 1 a vložený podformulář je typu n. Příklad podformuláře je na obrázku 1.5.





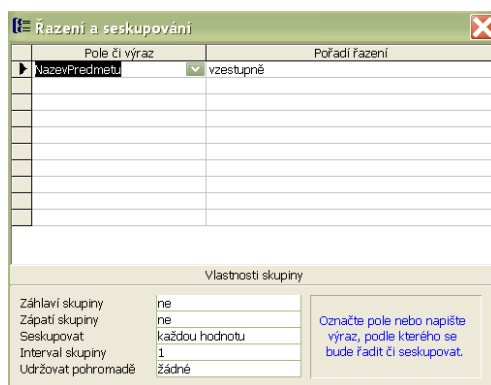
Obr. 1.5: Ukázka podformuláře v MS Access

## Sestavy

Sestavy jsou primárně určeny k tisku. Jako zdroj používají data z tabulek. Údaje v sestavě lze libovolně třídit, sumarizovat, seskupovat (až deset úrovní), vytvářet agregovaná data, uspořádaná podle určitých kritérií apod.

Ke každé vytvořené skupině lze zobrazit záhlaví a zápatí. Tato vlastnost je velmi důležitá při tvorbě agregovaných dat pro jednotlivé skupiny. Záhlaví a zápatí lze zobrazit také za jednotlivé stránky i za celou sestavu. Do sestav lze vkládat obrázky, kontingenční grafy i další podsestavy.

Jedna z nejdůležitějších funkcí při tvorbě sestav je **řazení a seskupování** (obrázek 1.6). Jak už bylo zmíněno, Access umožňuje řadit a seskupovat maximálně deset polí či výrazů.

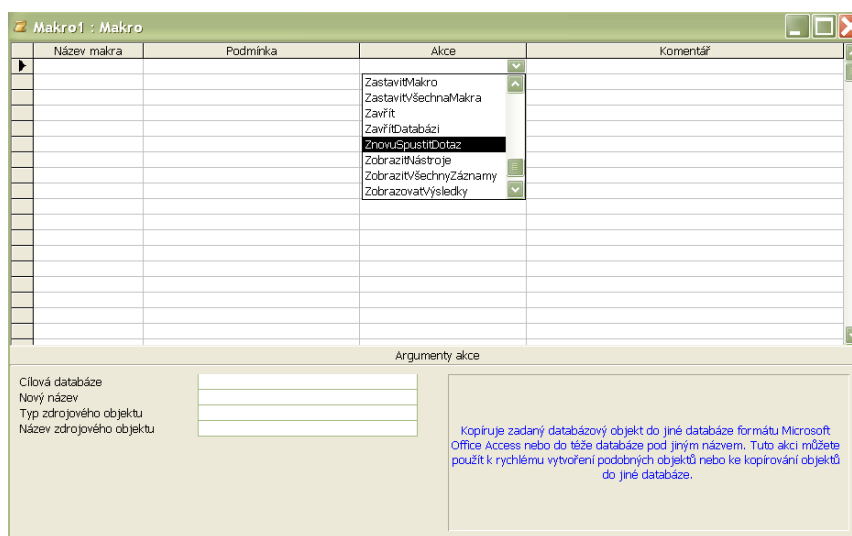


Obr. 1.6: Řazení a seskupování v MS Access

Za jednotlivé skupiny se používají tzv. agregační funkce. K dispozici jsou například tyto: *Sum*, *Avg*, *Min*, *Count*. Mimoto je možné vytvořit si vlastní funkci pomocí Visual Basicu.

## Makra

Makra v aplikaci provádějí operace, definované v tzv. návrhovém okně makra (obr. 1.7). V něm lze stanovit také podmínky, které určí jestli makro vůbec proběhne. MS Access má k dispozici 56 různých akcí, které mohou makra provádět. Tyto akce jsou logicky uspořádané do funkčních kategorií (např. testování podmínek a řízení toku akcí, nebo informování uživatele o činnostech apod.).



Obr. 1.7: Návrhové okno makra v MS Access

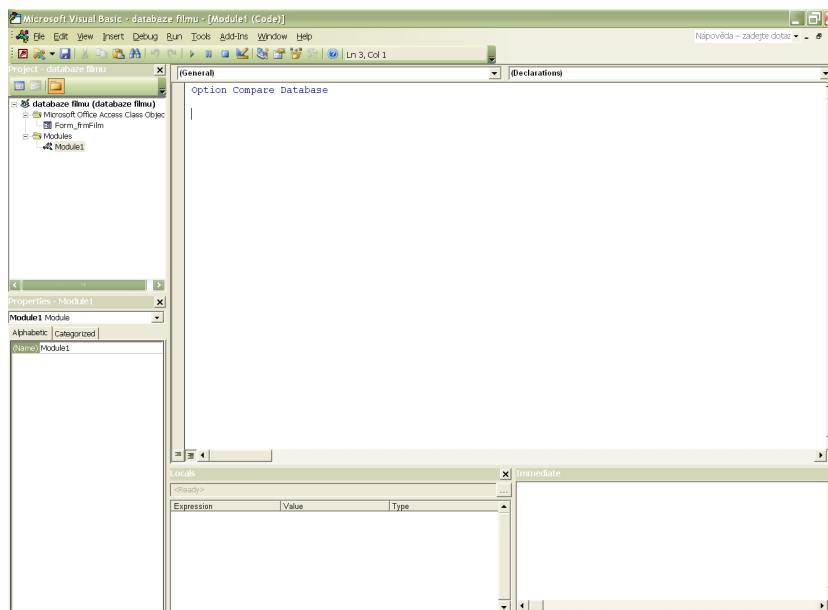
U každé akce lze nastavit specifické vlastnosti. Access umožňuje také převod vytvořeného makra do jazyka Visual Basic. Avšak provázání na formuláře a sestavy se musí upravit „ručně“.

## Moduly

Moduly jsou objekty napsané v jazyce Visual Basic. Narozdíl od maker je možné u modulů ošetřit možný výskyt chyb. VB je v Accessu plnohodnotný jazyk, proto se používá pro komplexnější řešení daného problému. Jakmile je aplikace hotová a veškerý kód ve Visual Basicu napsaný, je vhodné tento kód zkompileovat. Procesy se tak zrychlí a optimalizují. Pro psaní modulů Access používá Visual Basic Editor, viz obrázek 1.8.

### 1.7.1 Upravení výsledné aplikace

Na závěr je třeba vytvořit vhodné pracovní prostředí pro běžného uživatele. A to úpravou panelu nástrojů, místních nabídek, popřípadě vytvořením vlastní nabídky.



Obr. 1.8: Visual Basic Editor pro tvorbu modulů

Také je vhodné nastavit parametry „Po spuštění“, které určují například, jak bude probíhat spuštění aplikace, jestli se spustí určitý modul, či makro a co se uživateli otevře jako první (např. hlavní přepínací formulář).

### Úpravy panelu nástrojů a vytvoření vlastní nabídky

V programu MS Access je možné vybrat jakoukoliv nabídku a následně ji upravit. Lze ji také zkopírovat a úpravy provádět na právě zkopírované nabídce. Možné je také vytvořit si prázdnou nabídku a do ní vkládat libovolná tlačítka a funkce.

### Úpravy místní nabídky

Existují tři typy místní nabídky: v sestavě, ve formuláři a ovládací prvek ve formuláři. Tyto nabídky lze upravovat zvlášť.

### Hlavní přepínací formulář

Pro spuštění aplikace uživatelem je vhodné vytvořit tzv. hlavní přepínací formulář. Používá se pro navigaci v databázi a většinou funguje jako jakýsi „směrovník“ s odkazy na další formuláře.

## **Nastavení vlastnosti „Po spuštění“**

Zde se nastaví co se bude dít, když uživatel spustí databázi. Jaké místní nabídky, či řádky nabídek se zobrazí, nebo například povolení / zakázání změn těchto nabídek.

## **Zabezpečení databáze**

MS Access umožňuje zabezpečit databázi z pohledu neoprávněného přístupu. Vstup do aplikace lze omezit na konkrétní uživatele, nebo celé skupiny. MS Access ukládá tyto informace do systémového souboru přímo na daném PC a do souboru samotné databáze (.mdb). Pro vytvořené uživatele a skupiny lze upravovat oprávnění ke konkrétním objektům (tabulky, sestavy, makra . . .). Na závěr lze celou databázi zašifrovat tak, aby se zamezil přístup i za pomoci speciálního softwaru.

## Kapitola 2

# Analýza a zhodnocení současného stavu řešené problematiky

Nominace studentů ze zahraniční univerzity probíhá dvěma cestami:

1. Zahraniční univerzita pošle e-mail se jmény a e-mailovými adresami studentů a pracovníce na oddělení zahraničních styků na Ekonomické fakultě tyto studenty zkontaktuje sama.
2. Zájemci o studium ze zahraničí sami pošlou Student Application form s kontaktními údaji.

Následně vyplní formulář tzv. Learning Agreement, ve kterém uvedou předměty, které chtějí na škole studovat.

Oba formuláře, tj. Student Application form i Learning Agreement se zakládají v papírové podobě.

Podle studentských Learning Agreementů sepíše pracovníce za přijaté studenty seznamy předmětů, o které je zájem a informuje příslušné pedagogy, kolik studentů chce absolvovat jejich předmět. S pedagogy proběhne domluva na datu a čase první schůzky daného předmětu. Následně jsou informováni studenti.

Pokud výuka probíhá formou konzultací, další schůzky si studenti domlouvají přímo s příslušným pedagogem. Probíhá-li výuka ve třídě společně s českými studenty, řídí se platným rozvrhem.

Veškeré dokumenty jsou zpracovávány v Microsoft Wordu a Microsoft Excelu a zde se i tisknou potřebné materiály pro studenty.

Všechny materiály existují v papírové podobě a jsou uloženy v kanceláři. Tyto dokumenty musí být pět let archivovány na Ekonomické fakultě a po uplynutí této doby se převáží do archivu v Ostravě Porubě.

# Kapitola 3

## Návrh racionalizovaného řešení

### 3.1 Sémantické modelování

#### 3.1.1 Identifikace vstupních datových požadavků

Identifikace vstupních datových požadavků proběhla formou rozhovoru s uživatelem. Vzhledem k současnému stavu věcí, vznikla potřeba racionalizovat procesy spojené s vedením údajů o zahraničních studentech.

Pracovnice udržující tyto záznamy potřebuje určitý druh databáze, který ji umožní vložit údaje o zahraničním studentovi: jméno, příjmení, e-mail, telefon, univerzitu, která ho vyslala, na který semestr (zimní / letní / oba) a na jaký program (Erasmus, CEEPUS, bilaterální dohoda, jiné). Později dostane přidělen tzv. kód studenta a pseudo-rodné číslo. Dále je třeba, aby se student mohl zapsat na předmět, který bude mít určité kredity. Jehož datum první schůzky stanoví vyučující pedagog. Po ukončení předmětu bude studentovi přidělena známka.

Další požadavek na databázi spočívá v tisku údajů, tzv. sestav:

- Seznam studentů zapsaných na předměty (za každý jeden předmět).
- Student a jeho předměty s daty prvních schůzek.
- Seznam studentů s e-mailem a jejich univerzitou.
- Seznam předmětů.

Pedagog domlouvá se studentem individuální konzultace, které pak vykazuje. V tomto formuláři je ovšem potřeba podpis pedagoga. Jelikož však na škole ještě nebyl zaveden elektronický podpis, tak je stále nutná papírová forma. Proto zavedení tohoto formuláře do databáze v současné době pozbývá smysl.

Dále je třeba vytvořit mechanismus umožňující archivaci studentů, kteří již nejsou aktivní, jejich předměty a kdy na škole studovali.

### 3.1.2 Specifikace datových objektů a jejich charakteristik

Z rozhovoru byly zjištěny následující objekty a jejich charakteristiky:

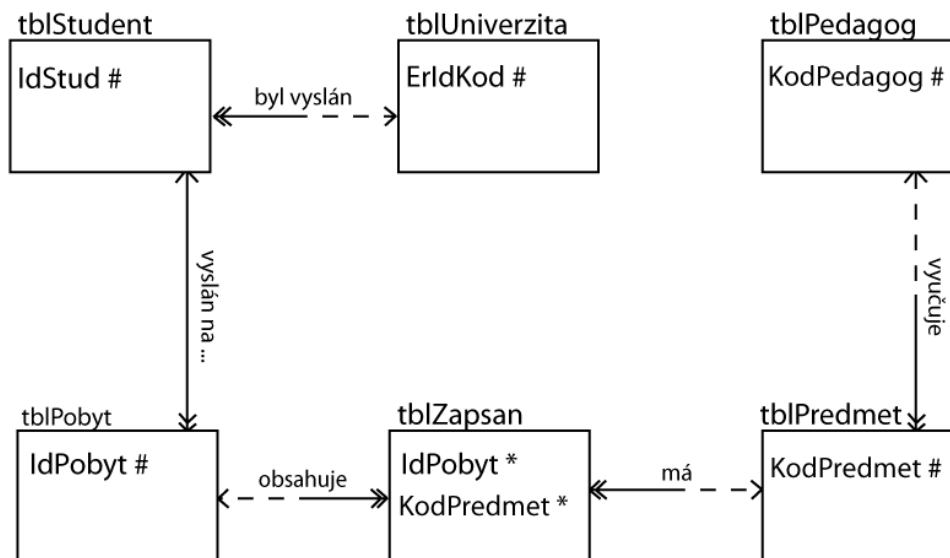
- Student — id kód, jméno, příjmení, e-mail, telefon, kód studenta, pseudo-rodné číslo.
- Univerzita — erasmus id kód, název univerzity.
- Pedagog — kód pedagoga, jméno, příjmení, email, telefon, číslo katedry.
- Předmět — kód předmětu, název předmětu, kredity, [datum první schůzky] . . . , [výsledek] . . . .
- Pobyt (program) — id pobytu, druh pobytu (Erasmus, . . . ), akademický rok, semestr, datum zahájení semestru, datum ukončení semestru.

### 3.1.3 Korekce struktury datových objektů + struktura entit pro konceptuální model

- Student — id kód (PK), jméno, příjmení, e-mail, telefon, kód studenta, pseudo-rodné číslo.
- Univerzita — erasmus id kód (PK), název univerzity.
- Pedagog — kód pedagoga (PK), jméno, příjmení, email, telefon, číslo katedry.
- Předmět — kód předmětu (PK), název předmětu, kredity.
- Pobyt (program) — id pobytu (PK), druh pobytu (Erasmus, . . . ), akademický rok, semestr, datum zahájení semestru, datum ukončení semestru.
- Zapsán — id pobytu (\*), kód předmětu (\*), datum první schůzky, výsledek.

## 3.2 Konceptuální modelování

Vymezení struktury entit bylo provedeno v E-R modelu, jak je patrné z obrázku 3.1.



Obr. 3.1: E-R model

### 3.3 Logické relační modelování

#### 3.3.1 Vytvoření soustavy předběžných relací

V této části byly určeny přesné názvy relací a k nim přiřazeny primární a cizí klíče.

- tblStudent (IdStud#, ErIdKod (c.k.))
- tblPedagog (KodPedagog#)
- tblPredmet (KodPredmet#, KodPedagog (c.k.))
- tblUniverzita (ErIdKod#)
- tblZapsan (IdPobyt \*, KodPredmet \*)
- tblPobyt (IdPobyt#, IdStud (c.k.))

#### 3.3.2 Přiřazení zbývajících atributů

K soustavě předběžných relací byly doplněny zbývající atributy.

- tblStudent (IdStud#, ErIdKod (c.k.), Prijmeni, Jmeno, Email, Telefon, Kod-Student, PseudoRodCis)
- tblPedagog (KodPedagog# PedPrijmeni, PedJmeno, PedEmail, PedTelefon, CisKat)



- tblPredmet (KodPredmet#, KodPedagog (c.k.), NazevPredmet, Kredity)
- tblUniverzita (ErIdKod#, NazevUniv)
- tblZapsan (IdPobyt \*, KodPredmet \*, PrvniSchuzka, Vysledek)
- tblPobyt (IdPobyt#, IdStud (c.k.), DruhPobytu, Semestr, AkadRok, ZacSem, KonecSem)

### 3.3.3 Specifikace domén

V tabulkách 3.1 až 3.6 jsou vysvětlivky k jednotlivým atributům.

V tabulce (tab. 3.7) jsou definovány domény nad jednotlivými atributy. Specifikuje se datový typ (text, číslo, datum, ...), velikost pole / formát, maska<sup>1</sup>, jedinečnost (zda-li jsou hodnoty unikátní) a null hodnoty (povolené / nepovolené).

#### Legenda k maskám:

- 0 — musí být zadáno číslo
- 9 — musí být zadáno číslo nebo nic
- L — musí být zadáno písmeno
- A — musí být zadáno písmeno nebo číslo
- "literál" — bude zobrazen obsah uvozovek

Student	tblStudent
Identifikační číslo	IdStud
Příjmení studenta	Prijmeni
Jméno studenta	Jmeno
E-mail studenta	Email
Telefon studenta	Telefon
Kód studenta	KodStudent
Pseudo-rodné číslo	PseudoRodCis

Tabulka 3.1: Vysvětlivky k relaci tblStudent

---

<sup>1</sup>V Accessu lze nastavit, zda-li se budou nebo nebudou ukládat literály společně s daty.

Pedagog	tblPedagog
Kód pedagoga	KodPedagog
Příjmení pedagoga	PedPrijmeni
Jméno pedagoga	PedJmeno
E-mail pedagoga	PedEmail
Telefon pedagoga	PedTelefon
Číslo katedry	CisKat

Tabulka 3.2: Vysvětlivky k relaci tblPedagog

Předmět	tblPredmet
Kód předmětu	KodPredmet
Název předmětu	NazevPredmet
Kredity předmětu	Kredity

Tabulka 3.3: Vysvětlivky k relaci tblPredmet

Zapsán	tblZapsan
Datum první schůzky	PrvniSchuzka
Výsledek absolvovaného předmětu	Vysledek

Tabulka 3.4: Vysvětlivky k relaci tblZapsan

Univerzita	tblUniverzita
Erasmus ID kód	ErIdKod
Název univerzity	NazevUniv

Tabulka 3.5: Vysvětlivky k relaci tblUniverzita

Pobyt	tblPobyt
Identifikační číslo pobytu	IdPobyt
Druh pobytu	DruhPobytu
Semestr	Semestr
Akademický rok	AkademRok
Datum začátku semestru	ZacSem
Datum konce semestru	KonecSem

Tabulka 3.6: Vysvětlivky k relaci tblPobyt

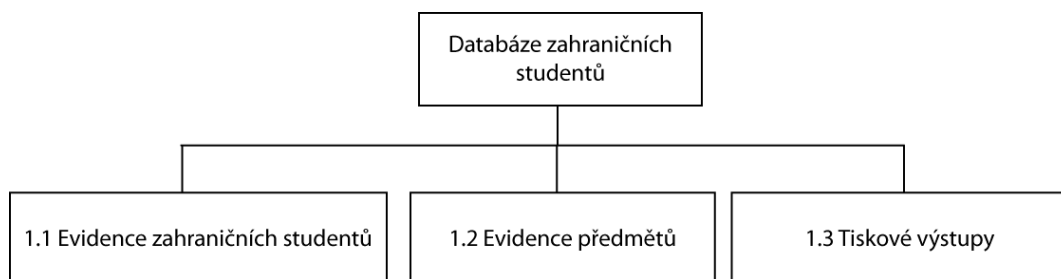
Atribut	Datový typ	Vel. pole / Formát	Maska	Jedineč.	Null
IdStud	automat. číslo	—	—	ano	ne
Prijmeni	text	50	—	ne	ne
Jmeno	text	40	—	ne	ne
Email	text	50	—	ne	ano
Telefon	text	13	”+”0000000000000	ne	ano
KodStudent	text	6	LLL000	ano	ano
PseudoRodCis	text	11	000000”/”00AA	ne	ano
KodPedagog	text	5	LLL00	ano	ne
PedPrijmeni	text	50	—	ne	ne
PedJmeno	text	40	—	ne	ne
PedEmail	text	50	—	ano	ano
PedTelefon	text	13	”+”0000000000000	ne	ano
CisKat	text	3	”1”00	ne	ne
KodPredmet	text	9	000000”/”00	ano	ne
NazevPredmet	text	60	—	ne	ne
Kredity	číslo	bajt	09	ne	ne
PrvniSchuzka	datum a čas	datum (obecné)	—	ne	ano
Vysledek	text	2	—	ne	ano
ErIdKod	text	15	—	ano	ne
NazevUniv	text	70	—	ne	ne
IdPobyt	automat. číslo	—	—	ano	ne
DruhPobytu	text	12	—	ne	ne
Semestr	text	5	—	ne	ne
AkademRok	text	9	0000”/”0000	ne	ne
ZacSem	datum a čas	datum (krátké)	—	ne	ne
KonecSem	datum a čas	datum (krátké)	—	ne	ne

Tabulka 3.7: Specifikace domén

## 3.4 Dekompozice funkcí

### Dekompozice systému: Databáze zahraničních studentů

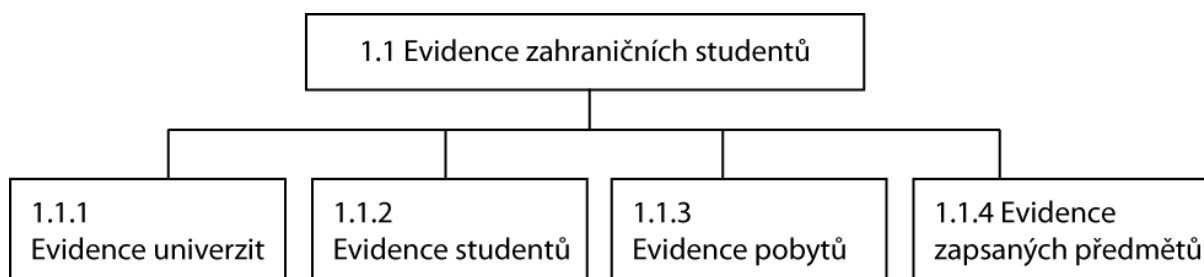
Na obrázku 3.2 je patrné základní členění funkcí v systému. Na evidenci studentů a na evidenci předmětů, které si přihlašují.



Obr. 3.2: Diagram hierarchické struktury funkcí

### Dekompozice funkce 1.1 Evidence zahraničních studentů (obr. 3.3)

Informace potřebné k evidování zahraničních studentů. Univerzita, která je vyslala, druh pobytu, díky kterému studují (Erasmus, CEEPUS, atd.) a předměty, které studenti chtějí absolvovat.



Obr. 3.3: Proces 1.1

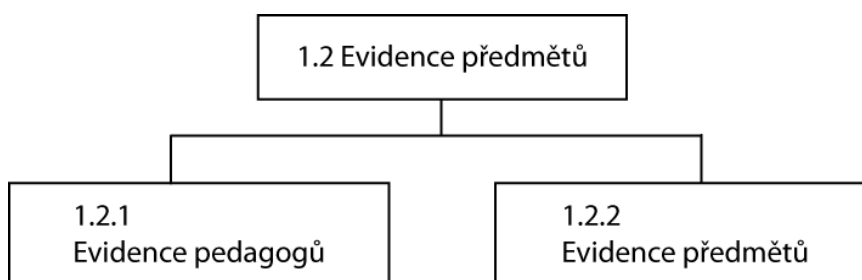
#### Popis funkcí:

- 1.1.1 Evidence univerzit — evidence všech nutných údajů pro vedení univerzity v databázi.
- 1.1.2 Evidence studentů — evidence všech potřebných informací týkajících se zahraničního studenta.

- 1.1.3 Evidence pobytů — evidence informací nutných pro vedení daného pobytu studenta.
- 1.1.4 Evidence zapsaných předmětů — zde se evidují informace o předmětech, které si student zapsal.

### Dekompozice funkce 1.2 Evidence předmětů (obr. 3.4)

Tato funkce umožňuje ukládání pedagogů a všech nutných informací o předmětech, které vyučují.



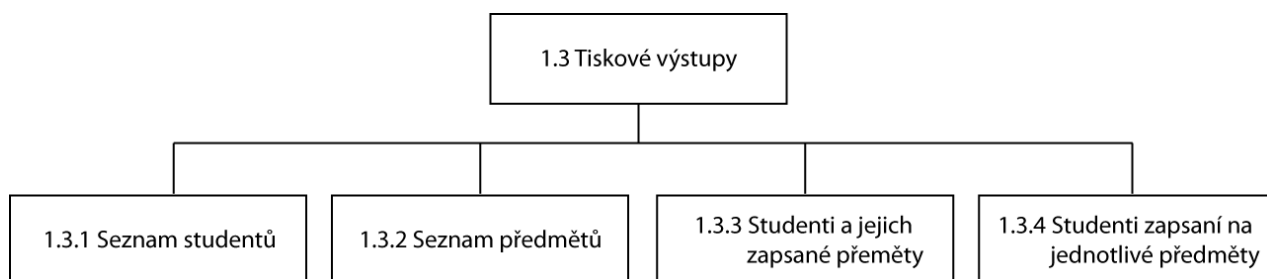
Obr. 3.4: Proces 1.2

### Popis funkcí:

- 1.2.1 Evidence pedagogů — vedení nezbytných údajů k evidenci pedagoga.
- 1.2.2 Evidence předmětů — evidování všech nutných informací týkajících se jednotlivých předmětů.

### Dekompozice funkce 1.3 Tiskové výstupy (obr. 3.5)

Funkce umožňuje tisk jednotlivých sestav.



Obr. 3.5: Proces 1.3

## Popis funkcí:

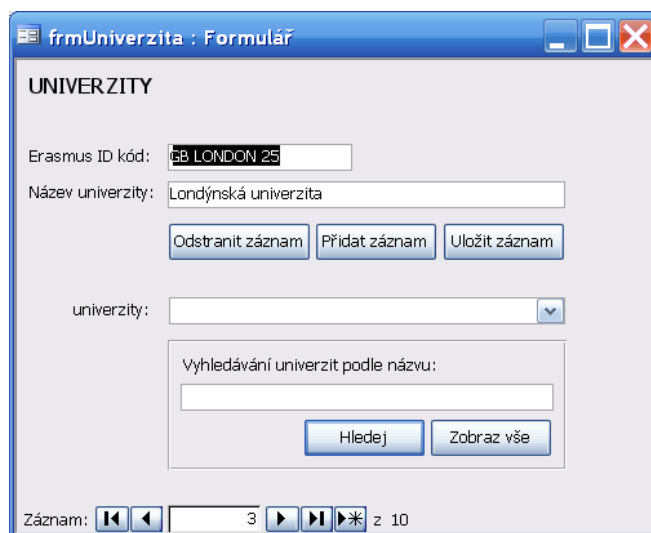
- 1.3.1 Seznam studentů — sestava obsahuje celé jméno, e-mail studenta a jeho domovskou univerzitu.
- 1.3.2 Seznam předmětů — sestava obsahuje název předmětu, jeho kód a kredity.
- 1.3.3 Studenti a jejich zapsané předměty — sestava vytiskne u každého studenta seznam jeho zapsaných předmětů s daty prvních schůzek.
- 1.3.4 Studenti zapsaní na jednotlivé předměty — vytiskne se seznam studentů zapsaných na jednotlivé předměty.

## 3.5 Procesní popis

### Funkce 1.1 Evidence zahraničních studentů

#### Funkce 1.1.1 Evidence univerzit

- Vstup: Informace o dané univerzitě.
- Proces: Uložení informací o univerzitě ve formuláři Univerzity (obr. 3.6).
- Výstup: Nový řádek v relaci tblUniverzita.



Obr. 3.6: Formulář Univerzity

### Funkce 1.1.2 Evidence studentů

- Vstup: Informace z tzv. Student Application form.
- Proces: Uložení informací o studentovi ve formuláři Evidence zahraničních studentů (obr. 3.7).
- Výstup: Nový řádek v relaci tblStudent.

The screenshot shows a web application window titled "frmPridaniStudenta : Formulář". At the top, there are three tabs: "Univerzity", "Pedagogové", and "Předměty". Below the tabs is a search section "Vyhledávání studentů podle příjmení:" with a text input field and two buttons: "Hledat" and "Zobraz vše". The main form is divided into two main sections: "STUDENT" and "POBYT STUDENTA". The "STUDENT" section contains fields for "ID studenta:" (with a dropdown for "matkové číslo"), "Příjmení:", "Jméno:", "E-mail:", "Telefon:", "Univerzita:" (with a dropdown), "Kód studenta:", and "Pseudo-rodné číslo:". There are buttons "Odstranit studenta", "Nový záznam", and "Uložit záznam" to the right. The "POBYT STUDENTA" section contains fields for "ID studenta:" (dropdown), "ID pobytu:" (dropdown for "matkové číslo"), "Druh pobytu:", "Semestr:", "Akademický rok:", "Začátek semestru:", and "Konec semestru:". There are buttons "Přidat záznam" and "Uložit záznam" below these fields. To the right of the "POBYT STUDENTA" section is a table with columns: "ID pobytu", "Kód předmětu", "První schůzka", and "Výsledek". Below the table is a button "Odstranit". At the bottom of the window, there are two sets of navigation controls. The first set shows "Záznam: 1 z 1" with navigation buttons. The second set shows "Záznam: 9 z 9" with navigation buttons. There are also buttons "Zpět na úvodní stránku" and "Ukončit aplikaci" at the bottom right.

Obr. 3.7: Formulář Evidence zahraničních studentů

### Funkce 1.1.3 Evidence pobytů

- Vstup: Informace týkající se pobytu daného studenta.
- Proces: Uložení těchto údajů ve formuláři Pobyt studenta (obr. 3.8).
- Výstup: Nový řádek v relaci tblPobyt.

### Funkce 1.1.4 Evidence zapsaných předmětů

- Vstup: Zapsání předmětu, který si student zvolil.
- Proces: Uložení informací o vybraném předmětu ve formuláři frmZapisovani-Predmetu (obr. 3.8).
- Výstup: Nový řádek v relaci tblZapsan

**POBYT STUDENTA**

ID studenta:

ID pobytu:

Druh pobytu:

Semestr:

Akademický rok:

Začátek semestru:

Konec semestru:

ID pobytu	Kód předmětu	První schůzka	Výsledek
1	123432/12	2.3.2009 10:45:00	<input type="button" value="Odstranit"/>
1			<input type="button" value="Odstranit"/>

Záznam:   1   z 8

Obr. 3.8: Formulář Pobyt studenta a frmZapisovaniPredmetu

## Funkce 1.2 Evidence předmětů

### Funkce 1.2.1 Evidence pedagogů

- Vstup: Nezbytné informace o pedagogovi.
- Proces: Uložení informací o pedagogovi ve formuláři Vyučující (obr. 3.9).
- Výstup: Nový řádek v relaci tblPedagog.

**VYUČJÍCÍ**

Kód pedagoga:

Příjmení:

Jméno:

E-mail:

Telefon:

Číslo katedry:

Vyhledat pedagoga podle příjmení:

Kód	Jméno	Příjmení
hal12	Halík	Adam
jer01	Jeroným	Alois
jos05	Josefák	Pepa
kos44	Kostka	Ludvík
mac02	Macek	Ladislav
ped11	Novák	Jaroslav
nov02	Nováková	Irena
nov11	Novotný	Jan
stu04	Štůr	Ludvík
vac22	Václavek	Josef

Záznam:   11   z 11

Obr. 3.9: Formulář Vyučující



## Funkce 1.2.2 Evidence předmětů

- Vstup: Informace o předmětu.
- Proces: Uložení informací o předmětu ve formuláři Předměty (obr. 3.10).
- Výstup: Nový řádek v relaci tblPredmet.

465766/45	Bakalářská práce - Databáze	15
321453/54	Bakalářský seminář	2
234567/89	Ekologie	2
123456/78	Ekonomie A	4
567543/45	Hospodářská politika	4
565543/24	Informatika A	3
123432/12	Logistika	3
346474/56	Mikroekonomie C	4

Obr. 3.10: Formulář Předměty

## Funkce 1.3 Tiskové výstupy

### Funkce 1.3.1 Seznam studentů

- Vstup: Informace z relací tblStudent a tblUniverzita.
- Proces: Aplikace vytvoří sestavu z dat uvedených relací.
- Výstup: Sestava rptStudent.

### Funkce 1.3.2 Seznam předmětů

- Vstup: Informace z relace tblPredmet.
- Proces: Aplikace vytvoří sestavu z dat uvedené relace.
- Výstup: Sestava rptPredmety.

### **Funkce 1.3.3 Studenti a jejich zapsané předměty**

- Vstup: Informace z relací tblStudent, tblPobyt, tblZapsan a tblPredmet.
- Proces: Aplikace vytvoří sestavu z dat uvedených relací.
- Výstup: Sestava rptStudentPredmety.

### **Funkce 1.3.4 Studenti zapsaní na jednotlivé předměty**

- Vstup: Informace z relací tblStudent, tblPobyt, tblZapsan a tblPredmet.
- Proces: Aplikace vytvoří sestavu z dat uvedených relací.
- Výstup: Sestava rptPredmetStudenti.

## Kapitola 4

# Zhodnocení výsledků zpracovaného řešení

Díky elektronickému zpracování dat pomocí databáze, lze výrazně ušetřit a zefektivnit procesy spojené s evidencí zahraničních studentů. Pracovnice proto již nepovede databázi v MS Excelu, kde mohlo docházet k nekonzistenci a redundanci dat a jiným potížím.

Práce se zpřehlední a urychlí. Bude možné vytvářet i různé statistiky. Díky základní vlastnosti databází, není třeba některá data opakovaně vkládat, např. jednou vložená univerzita lze použít pro více studentů; pedagog může vyučovat více předmětů, ovšem údaje o něm se do databáze vloží jen jednou, apod.

Zvlášť výrazným posunem je tisknutí sestav, které jsou nastavené podle potřeb uživatele. V případě nutnosti je lze upravit, tak aby plně splňovaly nové požadavky. V těchto sestavách se tisknou agregované údaje o studentech zapsaných na předměty; studenti a jejich zapsané předměty; seznam předmětů a seznam studentů.

Aplikace umožňuje archivaci vybraných údajů, tak aby je bylo možné zpětně dohledat.

Se zavedením elektronického podpisu na škole bude možné zrušit papírovou podobu vykazování individuálních konzultací pedagogy a přejít na elektronickou podobu.

V případě potřeby je možné vytvořenou databázi zahraničních studentů rozšířit o další části, nebo funkce, popř. ji modifikovat podle potřeb uživatele.

# Kapitola 5

## Závěr

Cílem mé bakalářské práce bylo navrhnout a vytvořit databázi zahraničních studentů. Samotná databáze byla vytvořena v programu Microsoft Access 2003.

V první kapitole jsou stručně rozebrána teoretická východiska řešení problematiky, základní vlastnosti databází a některé další důležité pojmy. Dále je pozornost zaměřena na tzv. tříúrovňovou koncepci datového modelování. Na závěr této kapitoly jsou popsány nejdůležitější aspekty programu MS Access 2003.

V druhé kapitole je analyzován momentální stav řešení problematiky, tj. jakým způsobem probíhá evidence zahraničních studentů v současné době.

Ve třetí kapitole je na základě analyzovaného stavu navrženo racionalizované řešení pomocí tříúrovňové koncepce datového modelování. Dále je znázorněna základní dekompozice funkcí a jsou zde krátce popsány procesy vkládání dat. Na základě datového modelu je vytvořena aplikace v MS Access 2003.

Jako zajímavá možnost se jeví formulář publikovaný na internetu, který by potenciální zahraniční student vyplnil a údaje odeslal do databáze.

Ve výhledu do budoucna lze uvažovat o různých modifikacích, popř. přebudování celého systému. V rámci zefektivnění práce s databází by bylo např. vhodné čerpat údaje o předmětech, popř. vyučujících přímo ze školního interního informačního systému. Toto řešení by však vyžadovalo hlubší prozkoumání a zvážení daných možností. Významnou roli by v tom hrála modernizace, popř. vytvoření nového IIS a možné zapojení databáze zahraničních studentů přímo do systému.

# Seznam použité literatury

- [1] KALUŽA, J. *Tvorba datového modelu v prostředí strategických informačních systémů*. Ostrava: Grafie, 1996. 115 s.
- [2] KALUŽA, J., KALUŽOVÁ, L., MAŇASOVÁ, Š. *Informatika*. 1. vyd. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2007. 167 s. ISBN 978-80-248-1293-9.
- [3] MOLNÁR, Z. *Automatizované informační systémy*. 1. vyd. Praha: Vydavatelství ČVUT, 2000. 126 s. ISBN 80-01-02269-2.
- [4] ŘEPA, V. *Analýza a návrh informačních systémů*. Praha: Ekopress, 1999. 403 s. ISBN 80-8611913-03.
- [5] VIESCAS, JOHN L. *Mistrovství světa v Microsoft Office Access 2003*. Přeložil Petr Matějů. 1. vyd. Brno: CP Books, a.s., 2005. 960 s. ISBN 80-251-0537-7.

# Seznam zkratek

- **BCNF** — Boyce–Coddova normální forma.
- **c.k.** — cizí klíč.
- **E–R model** — Entity–Relationship model.
- **IIS** — Interní Informační Systém
- **MS** — Microsoft.
- **ODBC** — Open Database Connectivity.
- **OLE** — Object Linking and Embedding.
- **RDBMS** — Relational Database Management System.
- **SQL** — Structured Query Language.
- **SŘBD** — Systémy řízení báze dat.
- **VB** — Visual Basic.
- **VBA** — Visual Basic for Applications.

# Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- byl(a) jsem seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. - autorský zákon, zejména § 35 - užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 - školní dílo,
- беру на ве́домі́, že Vysoká škola báňská - Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně ke své vnitřní potřebě diplomovou (bakalářskou) práci užít (§ 35 odst. 3),
- souhlasím s tím, že jeden výtisk diplomové (bakalářské) práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že údaje o diplomové (bakalářské) práci, obsažené v Záznamu o závěrečné práci, umístěném v příloze mé diplomové (bakalářské) práce, budou zveřejněny v informačním systému VŠB-TUO,
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona,
- bylo sjednáno, že užít své dílo — diplomovou (bakalářskou) práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne .....

Jméno a příjmení studenta

Adresa trvalého pobytu studenta: .....